



1 Abstract

“WinterLift” è un servizio su scala nazionale che si occupa di gestire e immagazzinare le informazioni relative alle aziende che gestiscono impianti di risalita in stazioni invernali e ai loro utilizzatori.

Usufruendo di esso i vari proprietari degli impianti (cabinovie o seggiovie) possono tenere traccia dello stato di quest'ultimi, registrando anche eventuali interventi di manutenzione, inoltre hanno la possibilità di raccogliere una vasta gamma di informazioni utili relative agli utenti che li frequentano, come le loro province di residenza oppure i loro vari passaggi negli impianti. Gli utenti infatti, per poter garantire un accesso agevole e sicuro agli impianti, dovranno essere dotati di una tessera sciatore univoca, la quale verrà scansionata ogni qual volta vorranno usufruire di uno degli impianti.

Con “WinterLift” quindi gli operatori degli impianti di risalita possono migliorare la loro pianificazione e gestione delle risorse, oltre a ottimizzare l'esperienza degli sciatori e personalizzare eventuali offerte basandosi sui dati raccolti. Gli utenti di “Winterlift” potranno oltretutto visualizzare tramite il servizio offerto i loro utilizzi dei vari impianti di risalita e avere informazioni in tempo reale sulla operatività degli impianti.

Questo rende “WinterLift” lo strumento ideale per ottimizzare l'efficienza operativa e offrire un'esperienza migliore possibile agli utenti degli impianti di risalita invernali.

2 Analisi dei requisiti

2.1 Descrizione testuale

Il progetto vuole mostrare una base di dati che tenga traccia non solo dei dati relativi agli impianti di risalita invernali ma anche dei dati relativi ai loro utilizzatori.

Dunque nella base dati troviamo innanzitutto le informazioni riguardanti gli **impianti**, di ogni impianto di risalita sono noti il **comprensorio** al quale appartiene, il codice che li identifica (unico nell'ambito del comprensorio), la **località** in cui è situato, l'altitudine alla partenza, l'altitudine all'arrivo ed i vari interventi di **manutenzione** al quale è stato sottoposto. Ci sono esattamente due diversi tipi di impianti: **seggiovia** e **cabinovia**. Di ogni seggiovia è noto il numero totale di sedili in dotazione, mentre di ogni **cabinovia** interessa il numero di funi. Ogni cabinovia trasporta delle cabine, di ogni **cabina** si rileva il numero di cabina (unico nell'ambito della cabinovia), la capienza (ovvero il numero massimo di persone che può trasportare la cabina), il numero di posti in piedi ed il suo peso.

Per ogni **comprensorio** sono noti il nome (identificativo), l'anno di inaugurazione e le aziende (almeno una) che lo gestiscono, ciascuna con la propria quota di investimento. Di ogni **località** interessa la provincia in cui si trova, il nome (unico nell'ambito della provincia), ed il numero di abitanti, mentre di ogni **provincia** si registrano il nome (identificativo), la superficie del territorio che ricopre e se è oppure no capoluogo di regione. Di ogni intervento di **manutenzione** sono noti l'impianto su cui è stato effettuato, la data di inizio e fine (è consentito iniziare un solo intervento al giorno per impianto ma ci possono essere più manutenzioni che si sovrappongono iniziate in date diverse, inoltre se la data di fine non è impostata la manutenzione è ancora in corso), il costo (preventivato nel caso in cui la manutenzione sia ancora in corso o finale altrimenti) e l'**azienda** che l'ha effettuato. Quando sono in corso uno o più interventi di manutenzione su un impianto questo diventa automaticamente non operativo. Di ogni **azienda** interessa la partita IVA (identificativo), il numero di dipendenti e la località in cui si trova la rispettiva sede principale.

Di ogni **persona** interessa il codice fiscale (identificativo), la data di nascita, il sesso e la provincia di residenza. Ogni persona può registrarsi ed ottenere la propria **tessera-sciatore** che dà il permesso di utilizzare gli impianti sciistici, ogni tessera sciatore contiene informazioni quali il proprietario di quest'ultima, il prezzo dell'abbonamento, dettagli assicurativi (testo semplice) e ha un identificativo univoco e una data di scadenza oltre il quale non è più possibile utilizzare gli impianti.

Infine viene registrato ogni passaggio delle persone (dotate di tessera-sciatore valida) sugli impianti di risalita. Di ogni **passaggio** interessano la data con l'ora in cui è avvenuto, l'impianto coinvolto e la persona che l'ha effettuato.

2.2 Glossario dei termini

Termine	Descrizione	Collegamenti
Impianto	Un impianto di risalita può essere una cabinovia o una seggiovia ma non entrambe, racchiude gli attributi comuni.	Passaggio, Manutenzione

Cabinovia	E' un tipo specifico di impianto di risalita, un sistema di trasporto a fune per persone su terreni montuosi, con cabine sospese su cavi, ideale per raggiungere luoghi panoramici o stazioni sciistiche	Entità figlia di Impianto, Cabina
Funi	Cavi adibiti al traino e sostegno delle cabine	Attributo di Cabinovia
Cabina	Cabina di una cabinovia	Cabinovia
Seggiovvia	E' un tipo specifico di impianto di risalita, un sistema di trasporto a fune utilizzato per il trasporto di persone su pendii o montagne. Consiste in una serie di sedili sospesi ad una corda	Entità figlia di Impianto
Comprensorio	Apprezzamento di terreno che può contenere degli impianti di risalita	Impianto,Azienda
Azienda	Azienda registrata nel sistema che può gestire dei comprensori e/o effettuare manutenzioni agli impianti	Manutenzione,Località
Quota	Investimento in percentuale di un'azienda su un comprensorio	Azienda, Comprensorio
Manutenzione	Manutenzione eseguita da un'azienda su un particolare impianto	Azienda,Impianto
Non operativo	L'impianto di risalita non è in funzione per via di interventi di manutenzione in corso	Impianto
Persona	Persona registrata nel sistema	Tessera-sciatore, Passaggio
Tessera-sciatore	Tessera che permette l'utilizzo degli impianti di risalita	Persona
Dettagli assicurativi	Testo semplice contenente informazioni riguardanti le coperture assicurative per infortuni o vari	Attributo di Tessera-sciatore
Passaggio	Il singolo utilizzo di un impianto da parte di una persona	Persona,Impianto
Provincia	Suddivisione amministrativa di un paese	Località
Località	Luogo specifico all'interno di una provincia	Azienda, Impianto, Provincia

2.3 Operazioni tipiche

Operazioni	Tipo (R/W)	Frequenza
Controllo della validità di una tessera sciatore (non scaduta)	R	10000 al giorno
Controllo operatività di un impianto di risalita	R	1000 al giorno
Inserimento nuovo passaggio	W	9000 al giorno
Inserimento nuova persona	W	800 a settimana
Inserimento nuova tessera sciatore	W	800 a settimana

Aggiornamento dati tessera sciatore	W	100 a settimana
Filtrazione passaggi per persona *	R	3000 al giorno
Controllo spese manutenzioni impianti	R	100 al mese
Valutazione profitto di un'azienda tramite interventi manutenzione	R	400 al mese
Inserimento nuovo intervento di manutenzione	W	30 a settimana
Chiusura intervento di manutenzione	W	30 a settimana

*Si assume che la filtrazione dei passaggi per persona avvenga con frequenza in ottica di un servizio di reportistica personale.

3 Progettazione Concettuale

3.1 Lista entità

Se non specificato, l'attributo è di default NOT NULL

- Provincia
 - Nome: VARCHAR(50) PRIMARY KEY,
 - Superficie: FLOAT, > 0
 - Is_capoluogo: BOOLEAN
- Località
 - Nome: VARCHAR(50) PRIMARY KEY,
 - Provincia: VARCHAR(50) PRIMARY KEY,
 - N_abitanti: INTEGER >= 0
- Comprensorio
 - Nome: VARCHAR(50) PRIMARY KEY,
 - Anno_inaugurazione: INTEGER
- Azienda
 - PIVA: CHAR(11) PRIMARY KEY,
 - Nome: VARCHAR(50),
 - N_dip: INTEGER >= 0
- Impianto
 - Codice: CHAR(4) PRIMARY KEY,
 - Comprensorio: VARCHAR(50) PRIMARY KEY,
 - Altitudine_partenza: INTEGER, > 0
 - Altitudine_arrivo: INTEGER, > 0
 - Tipo: ENUM = {'Seggiovia', 'Cabinovia'},
 - Is_operativo: BOOLEAN

L'entità impianto si specializza con una generalizzazione totale in:

- Cabinovia
 - Impianto: CHAR(4) PRIMARY KEY,
 - Comprensorio: VARCHAR(50) PRIMARY KEY,
 - N_funi: INTEGER >= 1

Inoltre ogni Cabinovia gestisce diverse cabine.

- Cabina
 - Cabinovia: CHAR(4) PRIMARY KEY,
 - Comprensorio: VARCHAR(50) PRIMARY KEY,
 - N_cabina: INTEGER PRIMARY KEY,
 - Capienza: INTEGER, >= 1
 - N_posti_in_piedi: INTEGER, >= 0
 - Peso: FLOAT >= 0

e:

- Seggiovia
 - Impianto: CHAR(4) PRIMARY KEY,
 - Comprensorio: VARCHAR(50) PRIMARY KEY,
 - N_sedili: INTEGER >= 1

- Manutenzione
 - Impianto: CHAR(4) PRIMARY KEY,
 - Compensorio: VARCHAR(50) PRIMARY KEY,
 - Data_inizio: DATE PRIMARY KEY,
 - Data_fine: DATE, (NULL se la manutenzione è ancora in corso, NOT NULL altrimenti)
 - Costo: FLOAT > 0
- Persona
 - CF: CHAR(16) PRIMARY KEY,
 - Data_nascita: DATE,
 - Sesso: Enum={'M','F'}
- Tessera_sciatore
 - ID_tessera: CHAR(12) PRIMARY KEY,
 - Data_scadenza: DATE,
 - Prezzo_abbonamento: FLOAT,
 - Dettagli_assicurativi: VARCHAR(256)
- Passaggio
 - Persona: CHAR(16) PRIMARY KEY,
 - Data_ora: TIMESTAMP PRIMARY KEY

3.2 Tabella delle relazioni

Relazione	Entità coinvolte	Descrizione	Attributi
Appartiene	Compensorio (0,N) Impianto (1,1)	Un compensorio può contenere zero o più impianti (poiché potrebbe essere sotto una gestione aziendale che in futuro vorrà creare un impianto su di esso) mentre un impianto appartiene ad uno ed un solo compensorio	Nessuno
Gestione	Compensorio (1,N) Azienda (0,N)	Un compensorio può essere sotto la gestione di una o più aziende, un'azienda può oppure no gestire più compensori	Quota: Float > 0
Sede	Azienda (1,1) Località (0,N)	Un'azienda ha una e una sola sede in una località mentre una località può essere sede di più aziende oppure no	Nessuno
Risiede	Località (1,1) Provincia (1,N)	Una località risiede in una e una sola provincia mentre una provincia contiene almeno una località	Nessuno
Risiede	Località (0,N) Impianto (1,1)	Una località può essere sede di nessuno o più impianti ma un impianto risiede in una e una sola località	Nessuno
Risiede	Provincia (0,N) Persona (1,1)	In una provincia possono risiedere zero o più persone ma una persona ha residenza in una e una sola provincia	Nessuno
Possiede	Persona (0,1) Tessera-sciatore (1,1)	Una persona può o meno avere una tessera-sciatore	Nessuno

		ma una tessera-sciatore deve appartenere a una ed una sola persona	
Esegue	Persona (0,N) Passaggio (1,1)	Una persona può eseguire zero o piu' passaggi su di un impianto di risalita mentre un passaggio viene eseguito da una ed una sola persona	Nessuno
Avviene	Impianto (0,N) Passaggio (1,1)	Un impianto può essere stato utilizzato per eseguire zero o più passaggi mentre un passaggio può avvenire in uno ed uno solo impianto	Nessuno
Effettua	Azienda (0,N) Manutenzione (1,1)	Una azienda può aver effettuato zero o più interventi di manutenzione mentre una manutenzione deve essere effettuata da una ed una sola azienda	Nessuno
Su	Manutenzione (1,1) Impianto (0,N)	Una manutenzione è eseguita su uno ed un solo impianto mentre un impianto può essere stato soggetto a zero o più manutenzioni	Nessuno
Appartiene	Cabinovia (1,N) Cabina (1,1)	Una cabinovia ha almeno una cabina mentre una cabina appartiene ad una e una sola cabinovia	Nessuno

3.3 Vincoli non rappresentabili tramite schema E-R:

- Una persona non può eseguire un passaggio se non possiede una tessera-sciatore valida, ovvero non scaduta.
- Una manutenzione non può avere data precedente alla data di inaugurazione del comprensorio su cui l'impianto sotto manutenzione è situato.
- Un impianto può essere o una cabinovia o una seggiovia ma non entrambe.
- Un passaggio in uno specifico impianto non può avvenire nei periodi di non operatività di quest'ultimo.


La violazione di uno qualsiasi dei vincoli illustrati viene notificata non concedendo l'inserimento/aggiornamento dei dati non validi tramite TRIGGER e FUNCTION.

Esempio (ultimo vincolo illustrato):

```
create or replace FUNCTION controllo_operativita_impianto_passaggio() RETURNS TRIGGER AS $$
BEGIN
    IF NEW.Impianto = (SELECT impianto from manutenzione M where M.impianto = NEW.impianto and M.comprendorio = NEW.comprendorio
                        and New.data_ora >= M.Data_inizio and (M.Data_fine IS NULL or M.Data_fine >= NEW.data_ora) group by Impianto)
    THEN
        RAISE EXCEPTION 'Passaggio non attuabile, impianto sotto manutenzione';
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

-- Creazione del trigger
CREATE TRIGGER check_operativita_impianto_su_passaggio
BEFORE INSERT OR UPDATE ON Passaggio
FOR EACH ROW
EXECUTE FUNCTION controllo_operativita_impianto_passaggio();
```

Quindi se si prova ad inserire un passaggio in un momento in cui l'impianto non è operativo si ottiene:

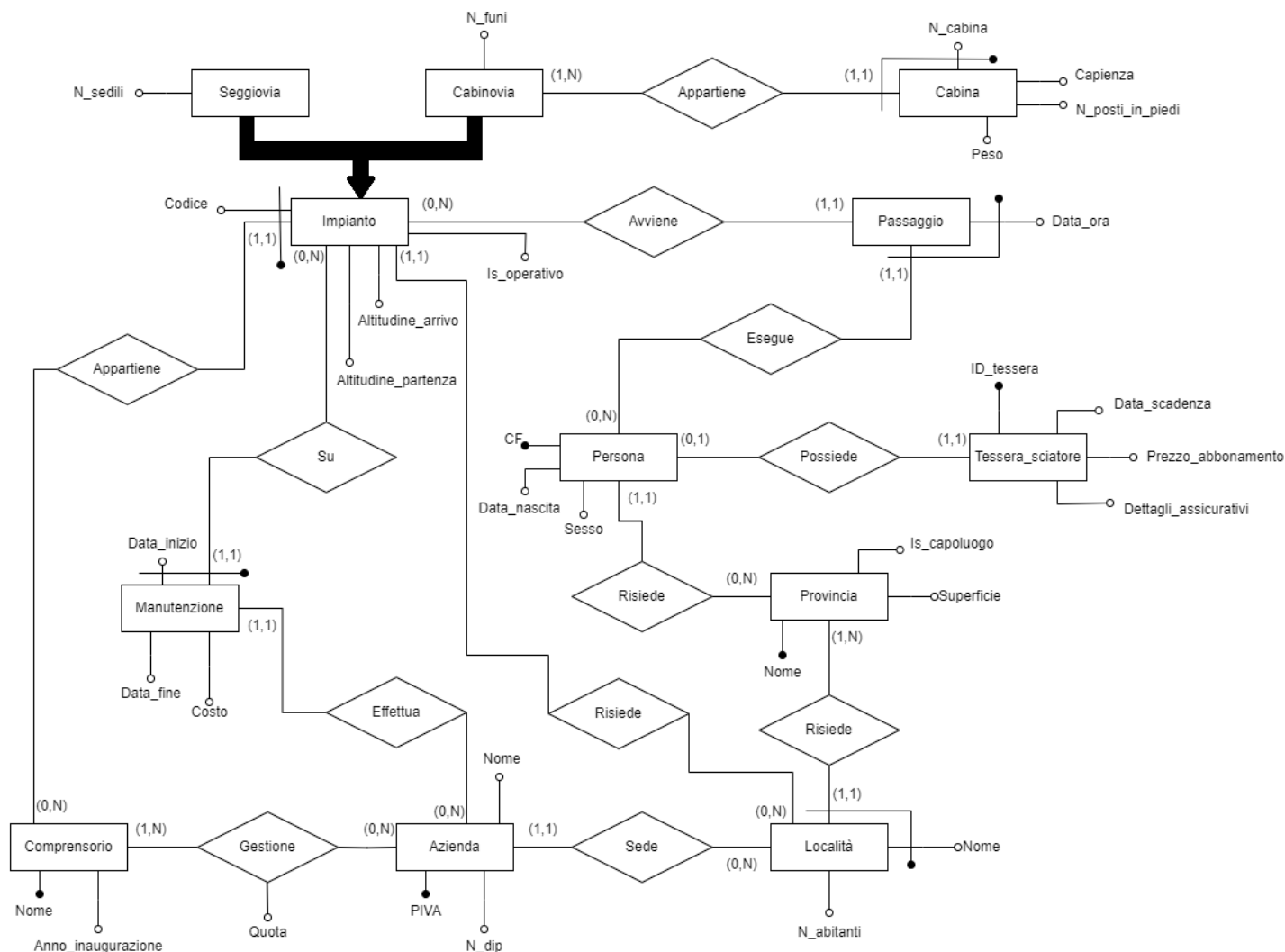
 ERRORE: Passaggio non attuabile, impianto sotto manutenzione CONTEXT: funzione PL/pgSQL controllo_operativita_impianto_passaggio() riga 6 a RAISE



3.4 Vincoli di derivazione:

- L'attributo *Is_operativo* dell'entità impianto è vero solo se non ci sono interventi di manutenzione relativi allo specifico impianto con attributo *Data_fine* non impostato, ovvero ancora in corso.

3.5 Schema Concettuale



4 Progettazione Logica

4.1 Ristrutturazione

4.1.1 Analisi delle ridondanze

L'attributo *Is_operativo* dell'entità *Impianto* è ridondante in quanto l'operatività o meno di un impianto può essere verificata controllando l'esistenza di un intervento di manutenzione sull'impianto stesso con l'attributo *Data_fine* non impostato che sta ad indicare una manutenzione ancora in corso.

Questa operazione viene resa rapida con l'aggiunta dell'indice "*manutenzione_impianto_index*" ([vedi 5.2](#)).

In conclusione l'attributo ridondante occuperebbe memoria senza portare alcun beneficio, con l'aggiunta del rischio di inconsistenza, quindi verrà eliminato.

4.1.2 Eliminazione delle generalizzazioni

Generalizzazione	Risoluzione
Impianto <= Cabinovia, Seggiovia	Per eliminare questa generalizzazione vengono mantenute le entità Cabinovia e Seggiovia poiché esse non rappresentano esattamente lo stesso concetto. Difatti l'entità Cabinovia ha relazioni e attributi differenti rispetto all'entità Seggiovia, non rendendo possibile accorpate le due entità in Impianto. Per poter fare ciò viene aggiunto l'attributo Tipo(ENUM ('Seggiovia', 'Cabinovia')) in Impianto che specifica se per l'impianto troviamo una relativa tupla in Seggiovia oppure in Cabinovia.

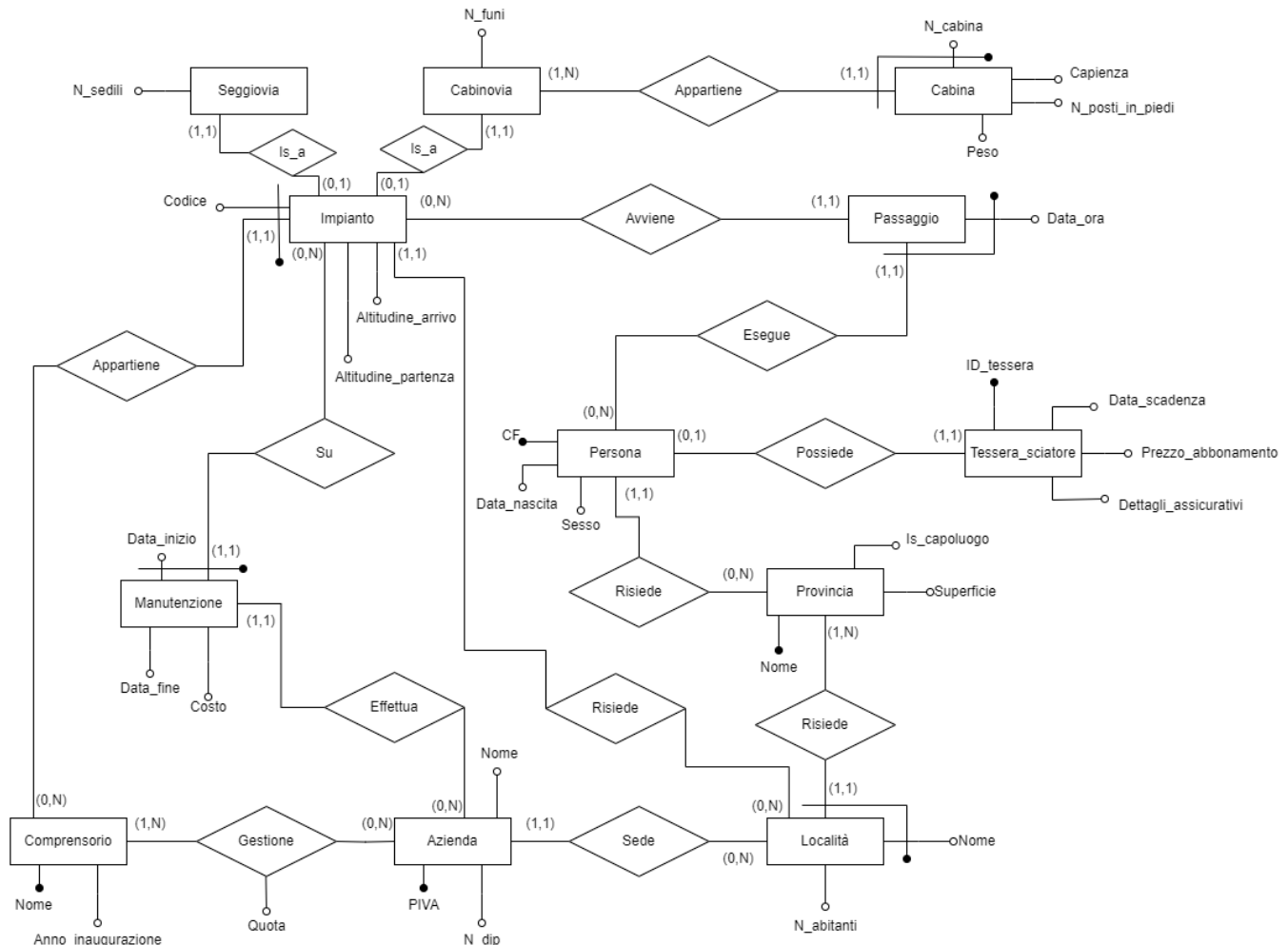
4.1.3 Scelta degli identificatori primari

Tra tutti gli identificatori è di rilievo un'analisi sulla scelta degli attributi primari delle entità **passaggio** e **manutenzione**.

Per l'entità **passaggio** è stato scelto come identificatore primario la Persona che lo effettua e la Data_ora (comprensiva di secondi) in cui avviene poiché non è possibile che una persona possa attuare più passaggi nello stesso istante.

Mentre per quanto riguarda l'entità **manutenzione** gli attributi Impianto, Compensorio e Data_inizio sono gli identificatori univoci poiché su uno specifico impianto non può iniziare più di una manutenzione al giorno.

Schema E-R ristrutturato



4.2 Creazione delle tabelle

Impianto(Codice, Compensorio, Altitudine_partenza, Altitudine_arrivo, Tipo, Localita_impianto, Provincia_impianto)

FK: Impianto.Compensorio => Compensorio.Nome,

Impianto(Localita_impianto,Provincia_impianto) => Localita(Nome,Provincia)

Cabinovia(Impianto, Compensorio, N_funi)

FK: Cabinovia(Impianto, Compensorio) => Impianto(Codice, Compensorio)

Cabina(Cabinovia, Compensorio, N_cabina, Capienza, N_posti_in_piedi, Peso)

FK :Cabina(Cabinovia, Compensorio) => Cabinovia(Impianto, Compensorio)

Seggiovia(Impianto, Compensorio, N_sedili)

FK: Seggiovia(Impianto, Compensorio) => Impianto(Codice, Compensorio)

Compensorio(Nome, Anno_inaugurazione)

Gestione (Azienda,Compensorio,Quota)

FK: Gestione.Azienda => Azienda.PIVA,

Gestione.Compensorio => Compensorio.Nome

Azienda(PIVA, Nome, N_dip, Localita_sede, Provincia_sede)

FK: Azienda(Localita_sede,Provincia_sede) => Localita(Nome,Provincia)

Manutenzione(Impianto, Compensorio, Data_inizio, Data_fine, Azienda, Costo)

FK: Manutenzione(Impianto, Compensorio) => Impianto(Codice, Compensorio)

Manutenzione(Azienda) => Azienda(PIVA)

Tessera_sciatore(ID_tessera, Data_scadenza, Prezzo_abbonamento, Dettagli_assicurativi, Proprietario)

FK: Tessera_sciatore.Proprietario => Persona.CF

Persona(CF, Data_nascita, Sesso, Residenza)

FK: Persona.Residenza => Provincia.Nome

Provincia(Nome, Superficie, Is_capoluogo)

Localita(Nome, Provincia, N_abitanti)

FK: Localita.Provincia => Provincia.Nome

Passaggio(Persona, Data_ora, Impianto, Compensorio)

FK: Passaggio(Impianto, Compensorio) => Impianto(Codice, Compensorio)

Passaggio.Persona => Persona.CF

5 Query e indici

5.1 Query

1. Una specifica azienda vuole avere un resoconto sugli interventi di manutenzione attuati sugli impianti.

Restituire per tale azienda il totale dei ricavi e dei giorni impiegati in interventi di manutenzione (la query dell'esempio utilizza "12345678901" come PIVA dell'azienda):

```
select
    azienda.piva,
    COALESCE(sum(costo), 0) as ricavo,
    COALESCE(sum(data_fine - data_inizio), 0) as giorni_lavoro
from
    azienda
left join manutenzione on manutenzione.azienda = azienda.piva
where
    azienda.piva = '12345678901'
group by
    azienda.piva
```

piva	ricavo	giorni_lavoro
12345678901	47200	15

2. Dopo un'analisi dei dati raccolti sugli impianti sciistici è emerso che la maggior parte dei frequentatori sono uomini, si è deciso dunque di inviare una promozione alle donne con tessera-sciatore in scadenza.

Mostrare il codice fiscale e l'id della tessera-sciatore delle donne la cui tessera scade tra meno di un mese :

```
select
    proprietario,
    ID_tessera
from
    tessera_sciatore
join persona on tessera_sciatore.proprietario = persona.CF
where
    sesso = 'F'
    and data_scadenza >= CURRENT_DATE
    and data_scadenza < CURRENT_DATE + interval '1 month';
```

proprietario	id_tessera
BNCLRA75C42D612A	TS002
RMNSFI94E43G675J	TS010
RSSMRC82A21E413P	TS011

Risultato con CURRENT_DATE ="2023-08-05"

3. Si vuole valutare quali impianti gestiti dal sistema sono stati penalizzati da interventi di manutenzione in uno specifico intervallo di tempo.

Stampare il codice, il comprensorio e il tipo degli impianti che sono stati non-operativi almeno un giorno all'interno di un determinato periodo (la query dell'esempio utilizza come intervallo di tempo il mese di dicembre 2022):

```
select distinct
    impianto.codice,
    impianto.comprensorio,
    impianto.tipo
from
    impianto
join manutenzione on impianto.codice = manutenzione.impianto
and impianto.comprensorio = manutenzione.comprensorio
where
    (manutenzione.Data_inizio >= '2022-12-01' :: date
and manutenzione.Data_inizio <= '2022-12-31' :: date) or
    (manutenzione.Data_fine >= '2022-12-01' :: date
and manutenzione.Data_fine <= '2022-12-31' :: date) or
    (manutenzione.Data_inizio < '2022-12-01' :: date
and
(manutenzione.Data_fine > '2022-12-31' :: date or manutenzione.Data_fine is null))
```

codice	comprensorio	tipo
0007	Comprensorio delle Alpi Apuane	Seggiovia
0008	Comprensorio dei Monti Sibillini	Cabinovia
0010	Comprensorio dei Monti Lattari	Seggiovia
0004	Comprensorio della Valle d Aosta	Seggiovia
0003	Comprensorio della Valtellina	Cabinovia
0006	Comprensorio delle Dolomiti	Cabinovia

4. Si vogliono ricavare informazioni sull'età dei frequentatori dei vari comprensori gestiti dal sistema.

Mostrare per ogni comprensorio, in ordine crescente, l'età media degli sciatori che hanno effettuato almeno un passaggio negli impianti di quest'ultimo:

```
drop view if exists frequentatori_comprensorio;

create view frequentatori_comprensorio as
select
    comprensorio,
    persona
from
    passaggio
group by
    (comprensorio, persona);

select
    comprensorio.nome,
    coalesce(
        TRUNC(
            avg(
                extract(
                    year
                    from
                    age(persona.data_nascita)
                )
            ),
            2
        ),
        null
    ) as eta_media
from
    comprensorio
left join frequentatori_comprensorio on frequentatori_comprensorio.comprensorio =
comprensorio.nome
left join persona on frequentatori_comprensorio.persona = persona.cf
group by
```

```
comprensorio.nome
order by
    eta_media asc;
```

nome	eta_media
Comprensorio dei Monti Lattari	28.00
Comprensorio del Sella Ronda	35.50
Comprensorio della Valle d Aosta	36.00
Comprensorio dei Monti Sibillini	37.00
Comprensorio del Cervino	38.00
Comprensorio delle Dolomiti	38.00
Comprensorio della Valtellina	43.00
Comprensorio della Val Gardena	null
Comprensorio dei Monti Lessini	null
Comprensorio delle Alpi Apuane	null

5.Si vogliono individuare gli impianti più frequentati gestiti dal sistema.

Selezionare il codice, il comprensorio e il numero di passaggi medi per giorno degli impianti che hanno un numero di passaggi medi giornalieri maggiore rispetto la media dei passaggi per giorno fra tutti gli impianti:

```
drop view if exists passaggi_per_giorno;

create view passaggi_per_giorno as
select
    passaggio.impianto,
    passaggio.comprensorio,
    date_trunc('day', passaggio.data_ora) as giorno,
    count(*) as n_passaggi
from
    passaggio
group by
    passaggio.impianto,
    passaggio.comprensorio,
    giorno;

select
    impianto,
    comprensorio,
    TRUNC(avg(n_passaggi),2) as n_passaggi_medi
from
    passaggi_per_giorno
group by
    impianto,
    comprensorio
having
    avg(n_passaggi) > (
        select
            avg(n_passaggi)
        from
            passaggi_per_giorno
    )
```

impianto	comprensorio	n_passaggi_medi
0003	Comprensorio della Valtellina	4.00
0002	Comprensorio del Sella Ronda	2.50
0001	Comprensorio del Cervino	6.00

6.Si vogliono individuare gli utenti affezionati ai soli impianti nella loro provincia di residenza.

Mostrare il codice fiscale e la provincia di residenza delle persone che hanno eseguito almeno un passaggio nei soli impianti situati nella loro provincia di residenza:

*Idea della query: Il numero di impianti differenti che una persona ha utilizzato deve essere uguale al numero di diversi impianti utilizzati da quest'ultima che sono situati nella sua provincia di residenza.

```
Select
    CF as codice_fiscale, Residenza
from
```

```

Persona Pe
join Passaggio Pa on Pa.Persona = Pe.CF
join Impianto I on Pa.impianto = I.Codice
and Pa.compensorio = I.compensorio
where
  I.Provincia_impianto = Pe.residenza
group by
  Pe.CF
having
  count(DISTINCT CONCAT(I.codice, '|', I.compensorio)) = (
    Select
      count(DISTINCT CONCAT(I2.codice, '|', I2.compensorio))
    from
      Persona Pe2
      join Passaggio Pa2 on Pe2.CF = Pa2.Persona
      join Impianto I2 on Pa2.impianto = I2.Codice
      and Pa2.compensorio = I2.compensorio
    where
      Pe2.CF = Pe.CF
  )

```

codice_fiscale	residenza
RSSGPP85D24L736I	Belluno
SPTMRA90E44G463P	Belluno

7. Si vogliono visualizzare gli impianti più capienti per ogni comprensorio.

Mostrare la cabinovia e la seggiovia per ogni comprensorio che possono trasportare più persone, tenendo conto del fatto che il numero di persone trasportate da una seggiovia è pari al numeri di sedili di quest'ultima mentre il numero delle persone trasportate da una cabinovia è pari alla somma della capienza di tutte le sue cabine:

*nel caso in cui più cabinovie/seggiovie possano trasportare lo stesso massimo numero di persone per comprensorio vengono esibite tutte.

```

select
  *
from
  (
    select
      'Seggiovia' as Tipologia,
      comprensorio,
      impianto,
      N_Sedili as posti
    from
      Seggiovia S1
    where
      N_Sedili = (
        select
          max(N_Sedili)
        from
          Seggiovia S2
        where
          S1.compensorio = S2.compensorio
        group by
          Compensorio
      )
    UNION
    select
      'Cabinovia ' Tipologia,
      C1.compensorio,
      impianto,
      SUM(capienza) as posti
    from
      Cabinovia C1
      join Cabina Cab1 on Cab1.Cabinovia = C1.impianto
      and Cab1.compensorio = C1.compensorio
    group by
      (C1.compensorio, impianto)
    having

```

```

SUM(Capienza) = (
    select
        max(posti_)
    from
        (
            select
                SUM(capienza) as posti_
            from
                Cabinovia C2
            join Cabina Cab2 on Cab2.Cabinovia = C2.impianto
            and Cab2.comprensorio = C2.comprensorio
            where
                C1.comprensorio = C2.comprensorio
            group by
                (C2.comprensorio, impianto)
            ) as Cabinovia_posti
        )
    ) as capienza_impianti_compensori
order by
    comprensorio

```

tipologia	comprensorio	impianto	posti
Cabinovia	Comprensorio dei Monti Lattari	0020	13
Seggiovia	Comprensorio dei Monti Lattari	0010	60
Cabinovia	Comprensorio dei Monti Lessini	0009	8
Seggiovia	Comprensorio dei Monti Lessini	0019	50
Seggiovia	Comprensorio dei Monti Sibillini	0018	80
Cabinovia	Comprensorio dei Monti Sibillini	0008	10
Cabinovia	Comprensorio del Cervino	0011	23
Seggiovia	Comprensorio del Cervino	0001	40
Seggiovia	Comprensorio del Sella Ronda	0012	40
Cabinovia	Comprensorio del Sella Ronda	0002	13
Cabinovia	Comprensorio della Val Gardena	0005	5
Seggiovia	Comprensorio della Val Gardena	0015	30

(Immagine ritagliata ai primi 12 risultati poiché troppo grande)

8. Si vogliono esporre ad ogni utente dei dati interessanti sui loro utilizzi degli impianti.

Per ogni persona restituire il numero totale di utilizzi degli impianti e la media del numero di passaggi per anno, tenendo conto per quest'ultima solo degli anni in cui è stato eseguito almeno un passaggio:

```

select
    persona_utilizzi.codice_fiscale,
    persona_utilizzi.numero_utilizzi,
    TRUNC(persona_media_utilizzi.media_pax,2) as media_passaggi_anno
from
    (
        select
            persona as codice_fiscale,
            count(*) as numero_utilizzi
        from
            passaggio PA
        group by
            PA.persona
    ) persona_utilizzi
join (
    select
        codice_fiscale,
        avg(pax_year) as media_pax
    from
        (
            select
                persona as codice_fiscale,
                count(*) as pax_year
            from
                passaggio PA
            group by
                (
                    PA.persona,
                    EXTRACT (
                        YEAR

```

```

FROM
    Data_ora
)
)
) as passaggi_anno_cf
group by
    codice_fiscale
) persona_media_utilizzi on persona_media_utilizzi.codice_fiscale =
persona_utilizzi.codice_fiscale

```

codice_fiscale	numero_utilizzi	media_passaggi_anno
CNTFNC88M57A319L	1	1.00
RMNSFI94E43G675J	1	1.00
RSSMRA80A01H501W	4	4.00
RSSGPP85D24L736I	11	3.66
RSSMRC82A21E413P	4	4.00
SPTMRA90E44G463P	1	1.00

9. Si vuole esibire un resoconto sugli interventi di manutenzione avvenuti sugli impianti. Per ogni impianto mostrare il tipo, il numero di manutenzioni fatte, il costo totale delle manutenzioni ed i giorni in totale non operativi:

```

select
    codice,
    I.compressorio,
    I.tipo,
    count(distinct Data_inizio),
    COALESCE(SUM(costo), 0) as costo_totale,
    COALESCE(SUM(Data_fine - Data_inizio), 0) as totale_giorni
from
    impianto I
    left join manutenzione M on M.impianto = I.codice
    and M.compressorio = I.compressorio
group by
    (I.codice, I.compressorio)

```

codice	compressorio	tipo	count	costo_totale	totale_giorni
0001	Compressorio del Cervino	Seggiovia	0	0	0
0002	Compressorio del Sella Ronda	Cabinovia	1	1000	5
0003	Compressorio della Valtellina	Cabinovia	1	500	2
0004	Compressorio della Valle d Aosta	Seggiovia	1	40000	3
0005	Compressorio della Val Gardena	Cabinovia	0	0	0
0006	Compressorio delle Dolomiti	Cabinovia	1	7000	10
0007	Compressorio delle Alpi Apuane	Seggiovia	2	3000	159
0008	Compressorio dei Monti Sibillini	Cabinovia	1	1200	5
0009	Compressorio dei Monti Lessini	Cabinovia	0	0	0
0010	Compressorio dei Monti Lattari	Seggiovia	3	7400	165

(Immagine ritagliata ai primi 10 risultati poiché troppo grande)

10. Una specifica azienda vuole avere un resoconto sulle spese totali di manutenzione che ha dovuto affrontare in seguito alla gestione dei compressori di cui possiede una quota.

Mostrare il totale delle spese dell'azienda pesato sulla quote di gestione di ciascun compressorio (la query dell'esempio utilizza "65432109876" come PIVA dell'azienda):

```

drop view if exists spese_gestione_per_aziende;

create view spese_gestione_per_aziende as
select
    compressorio.nome,
    gestione.azienda,
    sum(manutenzione.costo) as costi_totali,
    (sum(manutenzione.costo) / 100) * gestione.quota as spesa_personale
from
    manutenzione
    join impianto on impianto.codice = manutenzione.impianto
    and impianto.compressorio = manutenzione.compressorio
    join compressorio on impianto.compressorio = compressorio.nome
    join gestione on compressorio.nome = gestione.compressorio

```

```
group by
    (comprensorio.nome, gestione.azienda, quota);

select
    Azienda.PIVA,
    COALESCE(sum(spesa_personale), 0) as spesa_totale_aziendale
from
    azienda
    left join spese_gestione_per_aziende on azienda.piva =
spese_gestione_per_aziende.azienda
where
    azienda.PIVA = '65432109876'
group by
    azienda.PIVA
```

piva	spesa_totale_aziendale
65432109876	460

5.2 Indici

Il controllo della validità di una tessera-sciatore è un'operazione molto frequente in lettura nella base di dati ([vedi 2.3 Operazioni tipiche](#)) mentre l'inserimento di una nuova tessera-sciatore è molto meno frequente. Quindi si decide, ipotizzando uno sviluppo su larga scala, di indicizzare l'attributo *Id_tessera* di tessera-sciatore nell'indice *tessera_sciatore_index*.

```
create index tessera_sciatore_index on Tessera_sciatore (Id_tessera);
```

Inoltre la valutazione dei ricavi tramite interventi di manutenzione di una specifica azienda è un'altra operazione di lettura molto frequente nella base di dati, questo perché un'azienda potrebbe gestire un proprio portale e ottenere informazioni da "WinterLift" per calcolare in tempo reale i ricavi aziendali ([vedi 2.3 Operazioni tipiche](#)), mentre l'inserimento di un nuovo intervento di manutenzione è meno frequente. Data questa valutazione si decide di indicizzare l'attributo *Azienda* della tabella *Manutenzione* nell'indice *ricavi_azienda_index*.

```
create index ricavi_azienda_index on Manutenzione(Azienda);
```

Successivamente un altro indice viene inserito nella tabella *manutenzione* sugli attributi *impianto* e *comprensorio* poiché la valutazione e l'analisi delle spese fatte nel tempo da un impianto di risalita sono operazioni frequenti in lettura come anche la verifica della operatività di un impianto di risalita mentre l'inserimento di una nuova manutenzione è meno frequente. Anche in questo caso quindi, ipotizzando uno sviluppo su larga scala, indicizzare gli attributi citati precedentemente della tabella *Manutenzione* è una buona idea.

```
create index manutenzione_impianto_index on Manutenzione(Impianto, Comprensorio);
```

Infine un ultimo indice candidato è nella tabella *Passaggio* sull'attributo *Persona*, dato che un futuro portale dello sciatore dovrà frequentemente filtrare tra tutti i passaggi solo quelli personali dell'utente. Ma l'inserimento di nuovi passaggi è anch'esso un'operazione molto frequente ed un indice nella tabella *Passaggio* dunque degraderebbe le prestazioni. Si decide di conseguenza di non aggiungere ulteriori indici.

6 Codice C++

6.1 Descrizione dell'utilizzo del codice

Il codice C++ per l'esecuzione delle query consiste in un unico file `.cpp`, che va compilato attraverso il comando `g++ Codice.cpp -L dependencies/lib -lpq -o Queries`.

Al fine del login con il database sono state definite le seguenti costanti all'inizio del programma, le quali vanno ridefinite in base alla configurazione del database importato nella macchina locale:

```
PG_HOST, PG_USER, PG_PASS, PG_PORT, PG_DB.
```

Prima di poter compilare ed eseguire il programma ci si deve assicurare che i file `libpq.dll` e `libpq.lib` siano presenti in `./dependencies/lib` ed inoltre che i file `libpq-fe.h`, `pg_consigt_ext.h` e `postgres_ext.h` si trovino in `./dependencies/include`, dove `./` è il percorso della directory che contiene il file `.cpp`.

Una volta verificato ciò per eseguire il codice basterà avviare l'eseguibile "Queries".

Il programma per prima cosa tenterà di accedere correttamente al database, se ci riuscirà allora mostrerà a schermo la lista delle query, identificate da un numero da 1 a 10, altrimenti mostrerà un messaggio di errore e si interromperà. Per eseguire una query occorre inserire da tastiera il numero della query scelta, mentre per terminare l'esecuzione del programma va inserito '0'.

Alcune query (la numero 1, 3 e 10) richiedono l'inserimento di alcuni parametri da parte dell'utente:

1 e 10. Viene mostrata la lista di tutte le partite iva delle aziende gestite dal sistema. L'utente deve inserire la posizione all'interno della tabella della partita iva che vuole scegliere tra quelle disponibili.

3. Il programma chiede di inserire una data iniziale (ad esempio 2023-12-01) ed una finale (ad esempio 2023-12-31).

6.2 Documentazione del codice

Funzioni utilizzate dal codice:

```
PGconn* dbConnect(const char* host, const char* user, const char* db, const char* pass, int port)
```

Ritorna una connessione al database utilizzando i parametri passati come credenziali di accesso. Se la connessione per qualche motivo non va a buon fine allora mostra un messaggio di errore e termina il programma.

```
PGresult* executeQuery(PGconn* conn, const char* query)
```

Esegue una query passata come stringa ritornando il relativo risultato. Se l'esecuzione per qualche motivo non va a buon fine allora mostra un messaggio di errore e termina il programma.

```
void printQuery(PGresult* result)
```

Stampa in output sotto forma di tabella il risultato *result* di una query. Sostituisce i valori true e false con le stringhe "si" e "no" ed il valore null con la stringa "null". La funzione è in grado di gestire la dimensione delle colonne automaticamente.

```
void printLine(int campi, int* maxChar)
```

Funzione ausiliaria di printQuery che stampa in output una riga di separazione per la tabella.

```
char* choosePIVA(PGconn* conn)
```

Funzione che viene utilizzata per mostrare l'elenco di partite iva gestite dal sistema per poterne selezionare una (viene usata nelle query 1 e 10).